

GIS 中长事务模型

陈 波^{1,3}, 周顺平^{2,3}, 万 波^{2,3}, 叶亚琴^{2,3}

1. 中国地质大学研究生院, 湖北武汉 430074

2. 中国地质大学信息工程学院, 湖北武汉 430074

3. 武汉中地数码科技有限公司, 湖北武汉 430074

摘要: 为了解决传统的使用基于关系型数据库管理系统(relational database management system, RDBMS)存储数据的 GIS 软件平台在需要长事务并发的应用过程中出现问题, 从 GIS 对长事务需求和一般对事务的处理方式遇到的问题入手, 将 GIS 数据的编辑过程归结为数据库状态的不断演变, 通过增加存储表格的手段, 提出了一套切实可行的长事务解决方案, 并给出了相关问题的算法说明. 传统的并发是在编辑之前对要素加锁, 在这个编辑过程中, 其他用户都不能编辑这个要素, 直到第一个用户完成编辑, 释放锁, 其他用户才能对这个要素进行修改. 这样存在 2 个问题: 多个用户不能同时编辑一个要素; 后面的用户对数据的修改会覆盖掉前面用户的编辑结果. 为了解决这些问题, 本方案用状态标识数据的改变, 通过对状态的控制较好地解决了长事务的并发问题, 使多个用户不仅可以同时访问相同的要素, 而且不同的用户的编辑结果可以分别保存, 互不影响. 该方案已应用到实践中, 取得了良好的效果.

关键词: GIS; 长事务; 并发控制.

中图分类号: P208; TP311

文章编号: 1000-2383(2006)05-0605-04

收稿日期: 2006-05-30

Long Transactions in GIS

CHEN Bo^{1,3}, ZHOU Shun-ping^{2,3}, WAN Bo^{2,3}, YE Ya-qin^{2,3}

1. Graduate School, China University of Geosciences, Wuhan 430074, China

2. Faculty of Information Engineering, China University of Geosciences, Wuhan 430074, China

3. Wuhan Zondy Cyber-Tech Co., Ltd., Wuhan 430074, China

Abstract: This paper offers a solution to the application of a GIS system based on relational database management system (RDBMS) in a given circumstance which requires a long transaction. The traditional method to long transaction is to add locks on data in order to let someone edit it exclusively. That is to say, data can not be edited by more than one person at the same time with the traditional method. The paper starts from the requirement from GIS to long transaction and the questions which went on when people deal the long transaction with the traditional method. A feasible solution is offered which treats a data change as a statement and mark changes in data by state ids, then uses two algorithms to control the statements. With this solution not only data can be edited by few persons synchronously but also the different changes made by different people can be saved respectively.

Key words: GIS; long transaction; concurrency control.

0 引言

随着空间信息技术的发展, 地理信息系统 (GIS) 在普及性、可信度、成熟度等方面都有了较大

的提升, 并逐步成为进行资源管理和配置的最佳技术和对资源进行分析、决策和利用的有效工具. 从 GIS 在我国的发展过程来看, GIS 已经从最初的单用户版发展到多用户版、再到分布式版. GIS 的运行

环境在不断的复杂化,人机交互越来越多,处理事务所需的时间越来越长,这使得第四代 GIS 平台对长事务及多用户并发提出了更高的要求.

本文针对这种需求,研究了长事务管理和并发控制的原理,提出了基于面向实体的空间数据模型的 GIS 对长事务管理和并发控制的实现方案,并最终在国家“863”项目——“面向网络海量空间信息的大型 GIS”中得到应用.实践证明该方案能有效地提高大型 GIS 平台的多用户并发的能力,并成为第四代 GIS 系统(吴信才,2004)支持版本管理、离线编辑、历史数据回溯的理论基础.

1 事务处理模型

1.1 GIS 长事务

数据库管理系统(DBMS)中的长事务(Lt)可以表示为如下二元组(Garcia and Salem,1987).

$$Lt = (T, \rightarrow),$$

其中, $T = \{Lt1st1, Lt1st2, \dots, Lt1stn\}$, 为组成该长事务的各个子事务,每个子事务都是普通事务; $\rightarrow$ 是 T 上的一个偏序,是 T 中各个子事务在执行时应遵循的顺序约束.每个普通事务在可串行化调度下执行时,可以看成是对数据库状态所施加的一个变换,使其由一个一致状态转变到另一个一致状态为止(Abraham *et al.*, 2003).

GIS的发展经历了从文件到支持 DBMS 的过程,因此数据库中事务的概念也自然的引入到 GIS 中.在 GIS 的应用中大多数时候是人机交互的方式编辑图层,即人介入到活动的事务中,从计算机的角度看,事务就成了长事务(Hector *et al.*, 2003).对应于 DBMS 中的长事务的概念,人们对于地理数据库中长时间的访问和编辑就是 GIS 长事务,而这个过程中的每一步可以串行化调度的编辑动作就是普通事务.

1.2 经典的 Sagas 事务模型 GIS 长事务

为了提高 DBMS 中长事务的并发能力,Garcia and Salem(1987)提出了 Sagas 事务模型:将 Lt 分成一系列的小事务,每个小事务都有自己的 ACID 属性,这个 Lt 就叫做 Saga 事务,所有的小事务都进行了提交,那么整个 Saga 事务才算提交,如果 Saga 事务执行了一部分小事务就取消了,那么必须对已经提交的小事务进行补偿.

1.3 地理数据库状态机

面向实体的空间数据模型是将地理实体信息存

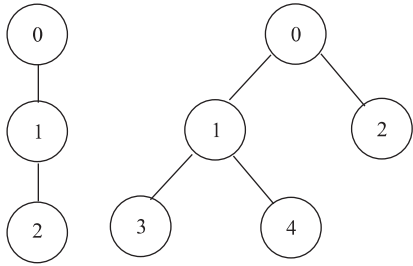


图 1 状态机的演变及状态分支

Fig. 1 Description of status change and parallel status branch

储在地理数据库(geodatabase, GDB)中,用要素类、对象类、注记类等管理这些信息.用户在 GDB 上对实体信息的编辑,就使 GDB 从一个状态转变到另外一个状态.

有穷状态机 M ,它是一个五元组: $M = \{S, \Sigma, l, s_0, F\}$; S 表示一组非空的状态, Σ 表示一组有限的输入, l 是一个映射 $S \times \Sigma \rightarrow P(S)$, 即一个状态转移到另外一种状态, $s_0 \in S$, 是初始状态, F 是一组状态机的结束状态,但是满足 $F \subseteq S$. 例如一个用户对 GDB 添加、更新、删除要素形成的 GIS 长事务可以用状态机表示,且多个用户从某个状态开始同时进行编辑,则状态机也可以产生分支,如图 1 所示,其中 $S = \{0, 1, 2\}$, $\Sigma = \{\text{“添加要素”、“更新要素”、“删除要素”}\}$.

2 解决方案

在图 1 表示的 GDB 的状态机中,能够发现如果事务顺序执行,状态机中的状态将线性的延伸下去,若用传统的事务处理方式即通过加锁来控制 GIS 中的长事务的执行,必然会存在以下问题:(1)多用户并发编辑过程中,多个用户的编辑结果会相互影响,用户无法独占自己的编辑结果;(2)用户在编辑时,可能长时间占用资源导致其他用户只能处于等待状态,并发程度不高.如果多个用户在同一状态下对 GDB 进行编辑的时候,会在原状态下产生不同的状态分支,这样就给多个用户同时编辑同一个要素提供了可能,但是还需要改变短事务处理的方式.

假设现在用户 A 和用户 B 同时编辑同一个要素类,并且假设用户每进行一次编辑都提交到数据库上.编辑之前的状态为 0.要素表(F 表)中主要的字段包括 FID(要素 ID);Data(要素的坐标信息).

编辑的初始状态和对应的存储如图 2.

2.1 添加要素

用户 A 添加一个要素后来到了状态 1, 用户 B 添加另外一个要素后来到了状态 2. 这时用户 A 和 B 希望看到的结果是已经存在的要素加上自己添加的要素, 但是又不希望看到其他用户编辑的结果.

如果添加的要素直接记录到 F 表中, 则 2 个用户都应该能够看到其他用户添加的要素. 因此用户添加的新要素应该用专门的表格进行存储, 这个表格拥有原数据表中的所有字段, 并且增加字段 stateID, 表示添加要素以后数据库所处的状态, 我们把这个专门存储新要素的表格叫做“添加表”, 简称“A 表”(图 3). 这样在指定状态下从 A 表中查找到要素加上 F 表中的要素, 即构成 A、B 用户各自期望的结果.

2.2 删除要素

用户 A 删除要素 100 来到了状态 1, 用户 B 删除要素 101 来到了状态 2. 同样的, 也不能让 A、B 用户对要素的删除直接反映到 F 表上, 所以也需要专门的

表格记录删除要素的操作. 这个表需要的字段包括: 删除要素的 ID, 该要素原来所处的 GDB 的状态和删除该要素以后 GDB 所处的状态. 称专门存储删除要素的表格为“删除表”, 简称“D 表”; 那么从 F 表中或 A 表中查到的要素再去掉 D 表中的要素, 剩下的即为用户能够看到的所有要素(图 4).

2.3 更新要素

用户 A 更新要素 100 来到了状态 1, 用户 B 更新要素 101 来到了状态 2. 有了 A 表和 D 表, 更新要素的操作可以理解为, 先在该状态下删除上个状态的那个要素, 再以相同的要素 ID 在 A 表中添加一个新要素, 也就是说, 更新要素需要在 A 表和 D 表中都添加记录.

这样一来, 将“数据库状态”、“A 表”和“D 表”的表示引入 GDB 以后, 用户编辑要素类时不再需要对 F 表加长事务的排他锁, 取而代之的是对 F 表的加共享锁和对 GDB 上相关的表加短事务的排他锁, 提高了长事务的并发性能, 进而解决了前面提出的 2 个问题: (1) 多用户并发编辑过程中, 用户在自己的状态分支上独占自己的编辑结果, 且不同状态分支上的结果不会相互影响; (2) 一个用户在编辑的时候, 其他用户也能马上开始自己的工作, 只是在该状态下产生了自己的状态分支.

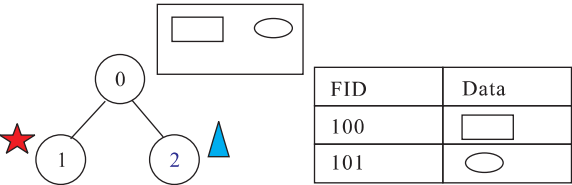


图 2 编辑前数据
Fig. 2 Data before edit

FID	Data	stateID
102		1
103		2

图 3 “添加表”
Fig. 3 “Append table”

FID	existStateID	stateID
100	0	1
103	0	2

图 4 “删除表”
Fig. 4 “Delete table”

3 算法

3.1 状态分支产生算法

- (1) 状态产生时, 如果父状态还没有产生其他子状态, 则产生的新状态的状态分支采用父状态的状态分支;
- (2) 状态产生时, 如果父状态已经产生了其他子状态, 则产生的新状态的状态分支用自己的状态号命名.

按照状态分支产生的算法, 图 1 中的并行状态中存在如下 3 个分支: 分支 1: (3、1、0); 分支 2: (2、0); 分支 3: (4、1、0).

3.2 查询算法

- 以要素的查询为例, 其算法如下:
- (1) 查询给定状态的状态分支号;
 - (2) 查询这个状态分支上包含的 GDB 状态列表;
 - (3) 查询所有 F 表中的要素, 过滤掉(2)中状态列表在中在 D 表中被删除的要素, 得到结果 1;
 - (4) 查询所有 A 表中的要素, 过滤掉(2)中状态

列表中在 D 表中被删除的要素,得到结果 2;
(5) 合并(3)、(4)中结果 1 和结果 2,得到最终查询结果.

4 结论

Sagas 事务模型强调执行补偿事务后 DBMS 要恢复到与以前状态等价的状态,即执行完补偿事务以后 DBMS 在 2 个状态下的差异是无法察觉的(Hector *et al.*, 2003). 本文实现的长事务处理方案在长事务需要回滚,即执行补偿事务时,只需简单的抛弃该长事务回滚之前产生的状态分支即可. 因此,这种长事务处理方式是支持 Sagas 事务模型的. 使用数据库状态标识数据变化的长事务解决方案,抛弃了传统事务处理过程中悲观加锁的方法,将长时间的锁转变成短时间的锁. 在事务处理过程中需要上“排他锁”的地方变化为上“共享锁”,从而大幅度地减少死锁的可能,提高了系统并发能力. 目前该方案已应用于 MAPGIS7.0 中,并成功地实现了对象类、要素类、注记类等的版本管理机制.

References

Abraham, S., Henry, E. K., Sudarshan, S., 2003. Database system concepts, fourth edition. Translated by Yang, D. Q., Tang, S. W.. China Machine Industry Press, Beijing (in Chinese).
Garcia, M. H., Salem, K., 1987. Sagas. In: Francisco, S., ed., Proceedings of the ACM SIGMDD conference. ACM Press, CA. 249—259.
Hector, G. M., Jeffrey, D. U., Jennifer, W., 2003. Database system; The complete book. Translated by Yue, L. H., Yang, D. Q., Gong, Y. C., et al.. China Machine Industry Press, Beijing (in Chinese).
Wu, X. C., 2004. The new generation of MAPGIS. *Geomatics World*, 2(2):3—7 (in Chinese with English abstract).

附中文参考文献

Abraham, S., Henry, E. K., Sudarshan, S., 2003. 数据库系统概念. 杨冬青, 唐世渭, 译. 北京: 机械工业出版社.
Hector, G. M., Jeffrey, D. U., Jennifer, W., 2003. 数据库系统全书. 岳丽华, 杨冬青, 龚育昌, 等, 译. 北京: 机械工业出版社.
吴信才, 2004. 新一代 MAPGIS. 地理信息世界, 2(2):3—7.